What is claimed is:

1. An apparatus comprising:

a trigger-response mechanism that includes at least one bank of user-programmable registers; and

a thread switch handler coupled to the trigger-response mechanism, the thread switch handler to invoke a second instruction stream responsive to an indication from the trigger-response mechanism that a user-defined trigger event has occurred during execution of a first instruction stream.

2. The apparatus of claim 1, wherein

the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that a synchronous user-defined trigger event has occurred during execution of the first instruction stream.

3. The apparatus of claim 1, wherein

the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that an asynchronous user-defined trigger event has occurred during execution of the first instruction stream.

4. The apparatus of claim 1, wherein

the thread switch handler is to save an instruction pointer address for the first instruction stream before invoking the second instruction stream.

5. The apparatus of claim 4, further comprising:

a task queue to receive the instruction pointer address.

6. The apparatus of claim 5, wherein:

the task queue further comprises a memory location.

7. The apparatus of claim 5, wherein:

the task queue further comprises a register.

8. The apparatus of claim 1, further comprising:

a plurality of event counters coupled to the trigger-response mechanism, wherein each event counter is to detect an atomic processor event.

9. The apparatus of claim 8, wherein

the thread switch handler is further to invoke the second instruction stream responsive to an indication from the trigger-response mechanism that an asynchronous user-defined trigger event has occurred during execution of the first instruction stream;

wherein the asynchronous user-defined trigger event is based on one or more of the atomic processor events.

10. The apparatus of claim 1, wherein

the thread switch handler is to save context information for the first instruction stream before invoking the second instruction stream.

11. The apparatus of claim 10, wherein:

the thread switch handler is further to save context for the first instruction stream in a memory location before invoking the second instruction stream.

12. The apparatus of claim 10, wherein:

the thread switch handler is further to save context for the first instruction stream in a register before invoking the second instruction stream.

13. The apparatus of claim 1, further comprising:

one or more user-programmable control registers coupled to the thread switch handler;

the value of the one or more control registers to indicate the weight of context information.

14. A system comprising:

a memory to store an instruction; and

a single-threaded processor coupled to the memory, wherein the processor provides a thread context;

wherein the processor includes a trigger-response mechanism to detect a user-specified trigger event and also includes a switch handler to invoke a helper thread responsive to occurrence of the trigger event.

15. The system of claim 14, wherein:

the memory is a DRAM.

16. The system of claim 14, wherein:

the instruction is a trigger instruction; and

the trigger-response mechanism is further to detect the opcode of the trigger instruction when the trigger instruction reaches an execution phase of an execution pipeline.

17. The system of claim 14, wherein:

the instruction is a marking instruction that specifies the trigger event, the trigger event being asynchronous; and

the trigger-response mechanism is further to detect the asynchronous trigger event.

18. The system of claim 14, wherein:

the switch handler is further to maintain minimal context information for a current thread before invoking the helper thread, wherein the minimal context information excludes traditional context information.

19. The system of claim 18, wherein:

the excluded traditional context information further comprises general register values.

20. The system of claim 18, wherein the minimal thread context information comprises an instruction pointer address value.

21. A method comprising:

detecting a user-specified trigger condition;

suspending execution of a first thread on a single-threaded processor;

utilizing hardware to save minimal context information for the current thread without operating system intervention; and

invoking a second thread on the single-threaded processor without operating system intervention.

22. The method of claim 21, wherein:

detecting a user-specified trigger condition further comprises determining that a trigger instruction has been encountered.

23. The method of claim 21, wherein:

detecting a user-specified trigger condition further comprises determining that an asynchronous condition specified in a marking instruction has been encountered.

24. The method of claim 21, wherein:

utilizing hardware to save minimal context information further comprises saving an instruction pointer address value.

25. The method of claim 21, further comprising:

determining that the first thread should be resumed ;

restoring the minimal context information for the first thread; and

resuming execution of the first thread without operating system intervention.

26. The method of claim 21, wherein detecting a user-specified trigger condition further comprises:

receiving a marker instruction that specifies the trigger condition; and

monitoring a plurality of atomic event indicators to detect the trigger condition.

27. The method of claim 21, wherein detecting a user-specified trigger condition further comprises:

generating an asynchronous response to indicate that the second thread should be invoked.